

Grammaires et analyse syntaxique



En bref

- › **Langue(s) d'enseignement:** Français
- › **Méthode d'enseignement:** En présence
- › **Organisation de l'enseignement:** Formation initiale
- › **Forme d'enseignement :** Cours magistral, Travaux dirigés & Travaux pratiques
- › **Ouvert aux étudiants en échange:** Non

Présentation

DESCRIPTION

Les grammaires algébriques sont utilisées en informatique pour définir une syntaxe structurée, par exemple la syntaxe d'un langage de programmation avec des constructions imbriquées comme des conditionnelles, boucles, définitions de fonctions et procédures. Ce module enseigne les grammaires algébriques d'abord d'un point de vue pratique : comprendre et pouvoir écrire des grammaires, constructions d'analyseurs grammaticales descendantes (technique LL1) et ascendantes (technique LR1), utilisation d'un générateur d'analyse grammaticale moderne (menhir), interaction avec l'analyse lexicale et construction d'un arbre de syntaxe abstraite. Pendant les dernières semaines, les aspects fondamentaux des grammaires seront étudiés, comme leur relation avec les automates à pile, et les limites d'expressivité des grammaires.

PRÉ-REQUIS NÉCESSAIRES

1. Cours Automates an Analyse Lexicale du L2 :

- * Expressions rationnelles
- * Automates déterministes, non-déterministes et avec epsilon-transitions
- * Détermination et élimination des epsilon-transitions
- * Lemme d'itération des langages rationnels

2. Cours Programmation Fonctionnelle du L3 :

- * Programmation fonctionnelle en OCaml
- * Programmer avec des structures de données algébriques (arbres) en OCaml
- * Utilisation basique des modules en OCaml

SYLLABUS

Sujets centraux

1. Rappel des expressions rationnelles et de l'analyse lexicale
2. Grammaires algébriques, dérivations et arbres de dérivation, grammaires ambiguës
3. Relation entre langages rationnels et langages algébriques
4. Grammaires LL(1) : calcul des ensembles FIRST et FOLLOW, transformation de grammaires en forme LL(1), programmation d'analyseurs descendants
5. Principe de l'analyse ascendante (*shift/reduce*)
6. Construction d'un automate caractéristique LR(0) ou LR(1), construction d'une table d'analyse LR(0) et LR(1)

Pour en savoir plus, rendez-vous sur > u-paris.fr/choisir-sa-formation

7. Utilisation d'un générateur d'analyse ascendante comme *menhir*.
8. Programmation de la chaîne d'analyse syntaxique jusqu'à la génération d'une syntaxe abstraite
9. Limitations des grammaires algébriques : le lemme d'itération des grammaires

Sujets optionnels

1. Formalismes équivalents aux grammaires (diagrammes de syntaxe, forme de Backus-Naur étendue)
2. Études de cas : analyse syntaxique de langages de programmation
3. Propriétés de clôture de la classe des langages algébriques
4. Automates à pile, traductions entre grammaires et automates à pile

INFORMATIONS COMPLÉMENTAIRES

Ce cours s'organise de la manière suivante :

- 2 h de CM (Cours Magistral) par semaine sur 12 semaines ;
- 2h de TD/TP (Travaux Dirigés et Travaux Pratiques) sur 12 semaines.

En bref

CONTACTS

Responsable pédagogique

Ralf Treinen

✉ Ralf.Treinen@irif.fr

Gestionnaire de Scolarité - Licence 3

Raja Moujid Taimés

☎ 01 57 27 68 93

✉ raja.taimés@informatique.univ-paris-diderot.fr

EN SAVOIR PLUS

🔗 <http://www.informatique.univ-paris-diderot.fr/formations/licences/accueil>

Pour en savoir plus, rendez-vous sur > u-paris.fr/choisir-sa-formation